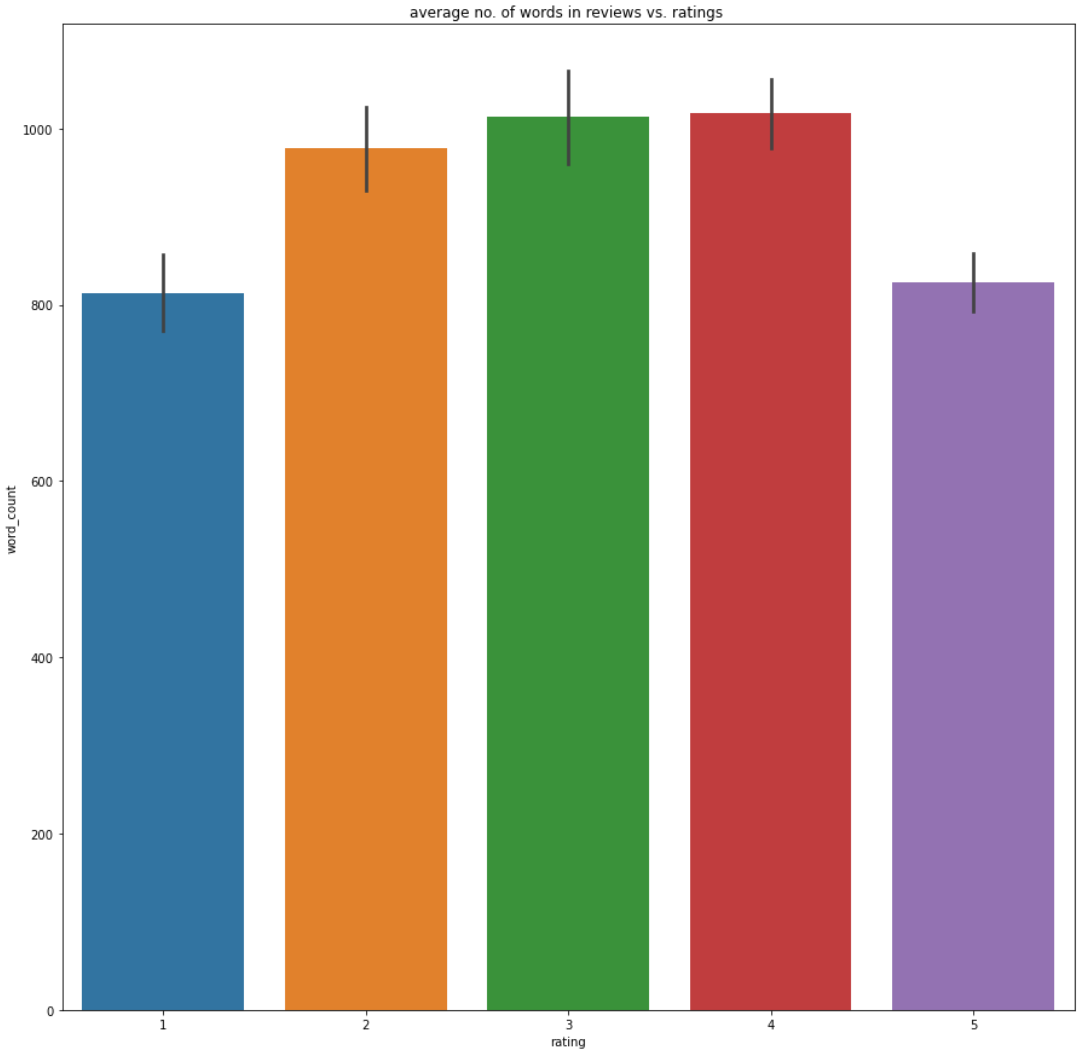
Problem statement

- ➢ Our target to build a model that detects customer feedback (like - dislike)
- ➢ I have worked on the dataset you have shared via email it was around 12k (preprocessed_kindle_review)

Data preparation

- ➢ Since it is an NLP application ,i focused only on rating and its corresponding reviewText
- ➢ Data was clean with non null values and no duplication (12000 cols and 4 rows)
- ➢ reviewText for each class almost have the same average number of words , which means that each class will have its distinguishing words or expressions

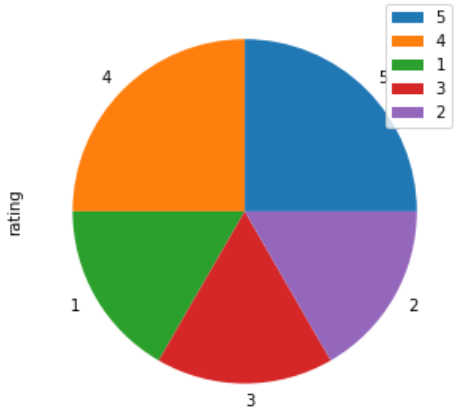


we can see

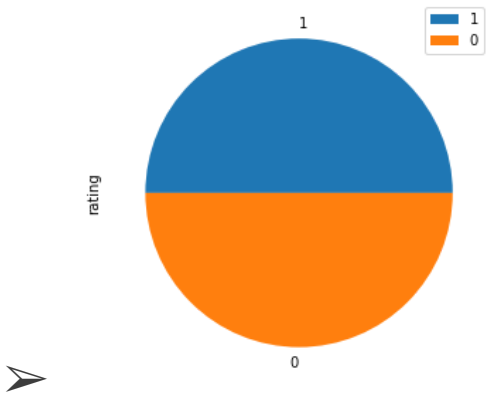
some correlated uni and bi gram with each class as follow

Label	Unigram	bigram
1	deleted waste	don waste waste time
2	loved didn	did finish finish book
3	ok okay	ok read just okay
4	liked enjoyed	good read enjoyed book
5	wonderful loved	wait read highly recommend

- Words and rating are making sense for sentiment analysis
- Rating percentage overview



- Rating values were from 1 to 5 , since our target was to detect the likes and not , i have mapped (1,2 ,3) to be negative feedbacks (dislike) and (4,5) to be positive feedback (like) , we will notice that after changing the rating to 0 and 1 , we will obtain balanced dataset with equal number of classes



Text cleaning

- As we will work on reviewText , from where our feature will be created
- The number of words per review is important thing to be checked , and important if it can be reduced
- Maximum number of words are 18715
- After building tfidf for features on this states and giving it parameters like stopping words and n gram 1 and 2 , i found that the features shape ((12000, 18945))
- After removing 2 ngram and only use unigram it was (12000, 9089)
- So for more reduction and by using NLTK library , wordnet lemmatizer , removing punctuation and english stop words , the reviewtext's number of words decreased and the maximum was 1134
- By checking head of our dataframe holding all reviews , we can clearly see the word reduction relative to reviewtext
-

	rating	reviewText	word_count	word_lemma	word_lemma_count
0	5	This book was the very first bookmobile book I...	**482**	book first bookmobile book buy school book clu...	**26**
1	1	When I read the description for this book, I c...	**3223**	read description book couldnt wait read downlo...	**181**
2	5	I just had to edit this review. This book is a...	**3772**	edit review book believe get right update rewr...	**238**
3	5	I don't normally buy 'mystery' novels because ...	**564**	dont normally buy mystery novels dont like how...	**34**
4	5	This isn't the kind of book I normally read, a...	**603**	isnt kind book normally read although try limi...	**40**

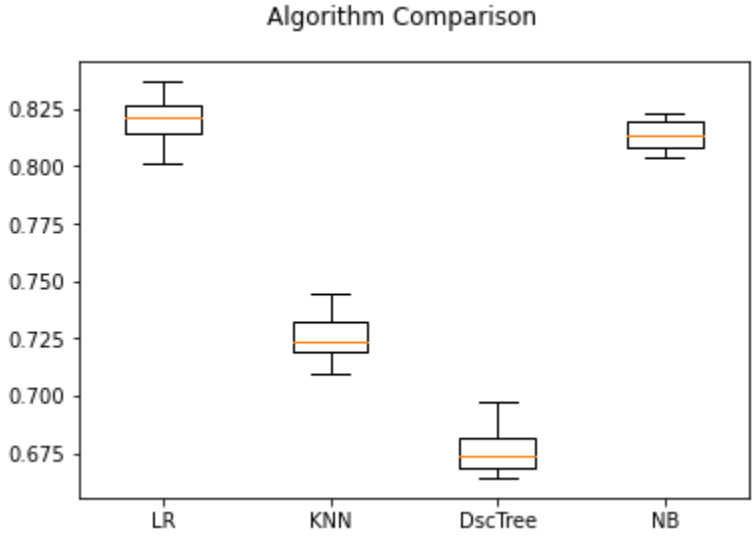
- After running TFIDF we have features with shape (12000, 7543) and this will be reflected on model building time

Building Model

I have tested 4 Machine learning models for text classification Logistic regression , decision tree , multinomial naive bayes and K neighbor classifiers

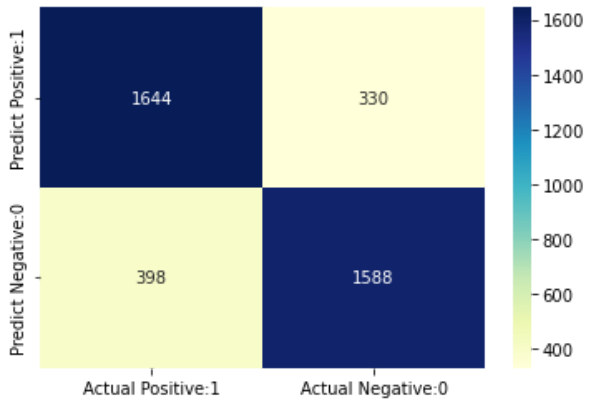
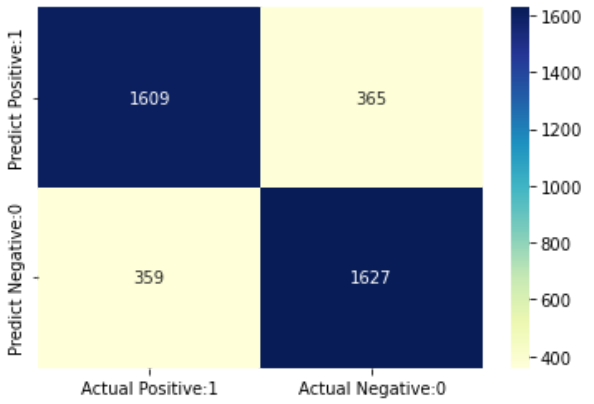
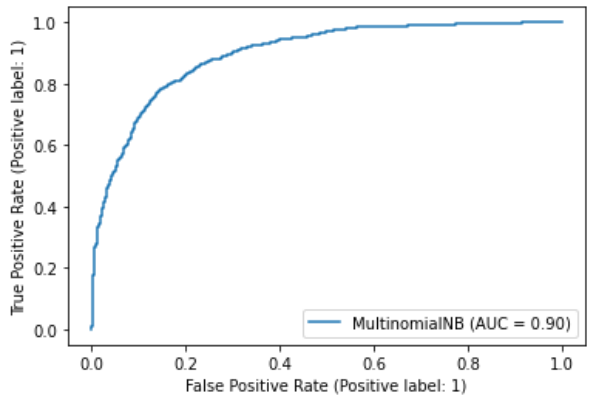
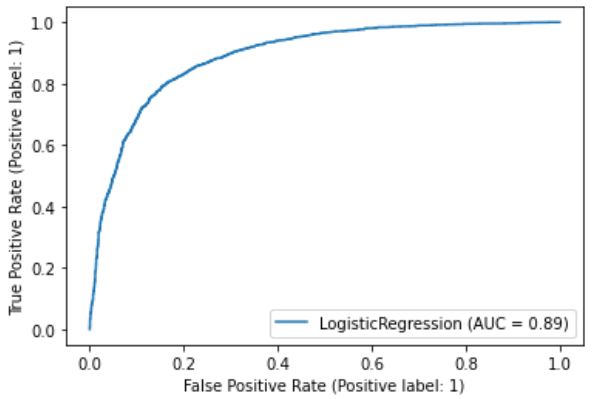
- Since data is balanced with equal number of class , classification accuracy will be used for measurement with cross validation
- confusion matrix and roc to decide the better model

Model	Cross validation score mean	Standard deviation
LR	**0.819417**	0.010814
KNN	0.725250	0.010382
DscTree	0.676000	0.010184
NB	**0.813250**	0.006795



- As shown from the analysis above , Logistic regression and NB almost the same accuracy average
- Will take a closer look on Logistic regression and Naive Bias models

➔ Confusion matrix and ROC

NB	LR
	
ROC NB	ROC LR
	

After analysis and comparing the above two models , they have little variance in accuracy , but NB with less standard variation and AUC is a little bit better than LR , will go for NB model
Also NB is faster than LR

ROC Interpretation¶

- ROC AUC is a single number summary of classifier performance. The higher the value, the better the classifier.
- ROC AUC of our model approaches towards 1. So, we can conclude that our classifier does a good job in predicting positive and negative feedbacks

Cross validation Interpretation

- Using the mean cross-validation, we can conclude that we expect the model to be around 81.3% accurate on average.
- If we look at all the 10 scores produced by the 10-fold cross-validation, we can also conclude that there is a relatively small variance in the accuracy between folds, So, we can conclude that the model is independent of the particular folds used for training.

Future work

A comparison with deep learning classifiers will be interesting with respect to the size of reviews to be added in consideration , i think with larger text , ML algorithms will not be enough